Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2**

з дисципліни: «Інформатика 2»

|  |  |
| --- | --- |
|  | Виконав : Першко Федір Сергійович  Група: РЕ-11  Викладачі: доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Підпис: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

**Мета роботи**: навчитися створювати програму для обчислення інтегралу чотирма методами.

**Ключові моменти**:

1. Вибір способу обчислення інтегралу:

do

{

printf("\nChoose the method of calculating:\n");

printf("\t1. By Left Rectangles method:\n");

printf("\t2. By Right Rectangles method:\n");

printf("\t3. By Trapezoid method:\n");

printf("\t4. By Simpson's method (parabola method):\n");

scanf("%u", &var);

if (var!=1 && var!=2 &&var!=3 &&var!=4)

printf("\nYou are mistaken\n");

}while (var!=1 && var!=2 && var!=3 && var!=4);

1. Обчислення інтегралу (на прикладі лівих прямокутників)

double left\_rectangle(double a, double b, double measurement\_error, unsigned int n)

{

double sum=0, x, h;

unsigned int i;

h = ( b - a ) / n;

x = a;

for (i = 0; i < n; i++ ){

sum += f(x);

x += h;

}

return sum\*h;

}

1. Підгін під задану похибку:

for (i = 0; i < 2; i++) {

if (i == 1){

n += 2;

I2 = left\_rectangle(a, b, measurement\_error, n);

}

if (fabs(I1 - I2) <= measurement\_error) {

break;

}

else {

i = 0;

}

}

**Код**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

double f( double x ) // основна функція

{

double y;

y = pow(x,2) + 2\*x;

//y = x;

return y;

}

double left\_rectangle(double a, double b, double measurement\_error, unsigned int n);

double right\_rectangle(double a, double b, double measurement\_error, unsigned int n);

double trap (double a, double b, double measurement\_error, unsigned int n);

double Simpson (double a, double b, double measurement\_error, unsigned int n);

int main()

{

double a, b, measurement\_error, I1, I2;

int n, var, i;

printf("\n\t\*===================================\*");

printf("\n\t|=\_\_\*-----------lab2------------\*\_\_=|");

printf("\n\t|=\_\_\*----------Integral---------\*\_\_=|");

printf("\n\t\*===================================\*\n");

printf("\n\tEnter the left boundary of integration\n X(first)=");

scanf("%lf", &a);

printf("\n\tEnter the right boundary of integration\n X(last)=");

scanf("%lf", &b);

do{

printf("\tEnter the number of partition intervals (N>0)\nN=");

scanf("%u", &n);

}while(n <= 0);

printf("\n\tEnter the measurment error of integration\n Measurment error=");

scanf("%lf", &measurement\_error);

do

{

printf("\nChoose the method of calculating:\n");

printf("\t1. By Left Rectangles method:\n");

printf("\t2. By Right Rectangles method:\n");

printf("\t3. By Trapezoid method:\n");

printf("\t4. By Simpson's method (parabola method):\n");

scanf("%u", &var);

if (var!=1 && var!=2 &&var!=3 &&var!=4)

printf("\nYou are mistaken\n");

}while (var!=1 && var!=2 && var!=3 && var!=4);

system("cls");

switch(var)

{

case 1:

{

printf("\n\n\t======\*Left Rectangles method\*======\n");

I1 = left\_rectangle(a, b, measurement\_error, n);

for (i = 0; i < 2; i++) {

if (i == 1){

n += 2;

I2 = left\_rectangle(a, b, measurement\_error, n);

}

if (fabs(I1 - I2) <= measurement\_error) {

break;

}

else {

i = 0;

}

}

} break;

case 2:

{

printf("\n\n\t======\*Right Rectangles method\*======\n");

I1 = right\_rectangle(a, b, measurement\_error, n);

for (i = 0; i < 2; i++) {

if (i == 1){

n += 2;

I2 = right\_rectangle(a, b, measurement\_error, n);

if (fabs(I1 - I2) <= measurement\_error) {

break;

}

}

else {

i = 0;

}

}

} break;

case 3:

{

printf("\n\n\t======\*Trapezoid method\*======\n");

I1 = trap(a, b, measurement\_error, n);

for (i = 0; i < 2; i++) {

if (i == 1){

n += 2;

I2 = trap(a, b, measurement\_error, n);

if (fabs(I1 - I2) <= measurement\_error) {

break;

}

}

else {

i = 0;

}

}

} break;

case 4:

{

printf("\n\n\t======\*Simpson's method (parabola method)\*======\n");

I1 = Simpson(a, b, measurement\_error, n);

for (i = 0; i < 2; i++) {

if (i == 1){

n += 2;

I2 = Simpson(a, b, measurement\_error, n);

if (fabs(I1 - I2) <= measurement\_error) {

break;

}

}

else {

i = 0;

}

}

}

}

printf("\n\ta = %.2lf \n\tb = %.2lf \n\tIntegral = %.8lf \n\tN = %d", a, b, I2, n);

return 0;

}

double left\_rectangle(double a, double b, double measurement\_error, unsigned int n)

{

double sum=0, x, h;

unsigned int i;

h = ( b - a ) / n;

x = a;

for (i = 0; i < n; i++ ){

sum += f(x);

x += h;

}

return sum\*h;

}

double right\_rectangle(double a, double b, double measurement\_error, unsigned int n)

{

double sum, x, h;

unsigned int i;

h = ( b - a ) / n;

x = a + h;

for (i = 0; i < n; i++ ){

sum += f(x);

x += h;

}

return sum\*h;

}

double trap (double a, double b, double measurement\_error, unsigned int n)

{

double sum, x, h;

unsigned int i;

h = ( b - a ) / n;

x = a;

for (i = 0; i < n; i++ ){

sum += ( f(x) + f( x + h ) ) / 2;

x += h;

}

return sum\*h;

}

double Simpson (double a, double b, double measurement\_error, unsigned int n)

{

double sum, sum1, sum2, x, h;

unsigned int i;

sum = f(a) + f(b);

h = ( b - a ) / n;

x = a;

for (i = 0; i < n; i++ ){

sum1 += f(x);

x += 2\*h;

}

x = a + h;

for (i = 1; i < n-1; i++ ){

sum2 += f(x);

x += 2\*h;

}

sum += 4\*sum1 + 2\*sum2;

return (h/3) \* sum;

}

